特設

第2章 標準規格ML-KEMと楕円曲線暗号のハイブリッド・アルゴリズムで

量子コンピュータでも解読が困難な耐量子暗号PQCを試す

古城 隆

表1 NISTが開催した暗号アルゴリズムPQCのコンペティション

時 期	内 容
第1ラウンド (2017年12月～2019年1月)	69の候補アルゴリズム
第2ラウンド (2019年1月～2020年7月)	69→26に候補を絞り込み
第3ラウンド (2020年7月～2022年7月)	26→最終候補7，代替候補8 選定(2022年7月) 鍵カプセル化 • CRYSTALS-Kyber(ML-KEM) ディジタル署名 • CRYSTALS-Dilithium (ML-DSA) • FALCON (標準化作業中) • SPHINCS+ (SLH-DSA)
第4ラウンド	選定(2025年3月) 鍵カプセル化 HQC (Hamming Quasi-Cyclic)

量子コンピューティングの脅威

量子コンピューティングは，明るい未来のために期待される技術である反面，その演算能力をセキュリティ上の攻撃に使われると，既存の暗号技術にとって深刻な脅威にもなります．現在，広く使われているRSA公開鍵暗号や楕円曲線暗号を量子コンピュータで簡単に解読してしまう手法も既に知られています．

そのような状況の中，NIST(米国国立標準技術研究所)は2017年，量子コンピュータでも解読が困難な暗号アルゴリズムであるPQC(Post-Quantum Cryptography，耐量子暗号)のコンペティションを開催しました(**表1**)．特に，公開鍵暗号とディジタル署名に関するアルゴリズムが対象となりました．世界中の暗号研究者から応募があり，2024年には，公開鍵暗号における鍵共有方式とディジタル署名に関する新たな標準規格が発行されています．

新しいアルゴリズムの研究開発は今でも続けられているのですが，標準化が完了しているアルゴリズムについては，TLSのようなセキュリティ・プロトコルの中で使えるようになりつつあります．

ここでは，PQCアルゴリズムを使ったTLSの鍵共有を簡単なPythonクライアントで実験します．PQCアルゴリズムの原理は従来型とは全く異なるものですが，TLSのようなセキュリティ・プロトコルでは全体のフレームワークを変更することなくこれを実装できます．

表2 PQCを使ったTLS通信で使用するファイル(第1章のTLSクライアントを一部改造)

分 類	ファイル名	内 容
テスト・プログラム	testTLS_PQC.py	PQCによるTLSハンドシェイクとアプリケーション・メッセージのテスト
TLSプロトコル	ClientHello.py	ClientHelloメッセージの送信
	ServerHello.py	ServerHelloメッセージの受信
	Finished.py	Finishedメッセージの送受信
	HandShakeMsg.py	ハンドシェイク・メッセージ，アプリケーション・メッセージの送受信
	TLSrecord.py	TLSレコードの送受信
	KeyExchange_PQC.py	鍵共有(含PQC)の処理
	KeySchedule.py	鍵スケジュールの処理
デバッグ	SslKeyLog.py	Wiresharkのための KeyLog生成

PQCを使ってTLS通信をさせてみる

本稿では，PQCの鍵共有に使う標準アルゴリズムML-KEM(Module-Lattice-Based Key-Encapsulation Mechanism)をTLSプロトコルで使ってみます(**表2**)．

プログラムの仕組みが分かるように，第1章で紹介したPythonによるTLSクライアントを使います．暗号アルゴリズムの中身についてはwolfSSLで提供されているML-KEMを，Pythonラッパのwolfcrypt-pyを通して使用します．

対向テストの相手となるTLSサーバは，wolfSSLのサンプル・サーバを利用します．動作環境は既に特設第1章で使った環境がダウンロード，セットアップされているものを使います．